ON THE COMPUTATION OF THE GENERALIZED INVERSE OF A POLYNOMIAL MATRIX

N. P. KARAMPETAKIS, P.TZEKIS

The main purpose of this note is to present a quicker and less expensive in memory algorithm for the generalized inversion of polynomial matrices than the existing ones presented in Karampetakis (1997a, 1997b)..

1 Introduction

Consider the polynomial matrix

$$A(s) = A_0 + A_1 s + \cdots + A_{q_1} s^{q_1} \in R[s]^{r\times m} \qquad (1)$$

where $A_i \in R^{r\times m}$, $i = 0, 1, ..., q_1$ where $r \neq m$ or $r = m$ and $\det[A(s)] = 0$. One of the most important numerical problems in linear system analysis and synthesis is the evaluation of the generalized inverse of $A(s)$. The reason of this interest is the large number of its implications in multivariable systems i.e. inverse systems (Lovass[8 9 10]), solutions of systems (Karampetakis[5]), solutions of matrix diophantine equations (Karampetakis[5]) which gives rise to numerous applications (Kucera[7]) etc. .

In case where $A(s) = A_0 \in R^{r\times m}$ the problem has been investigated by Penrose[11] and a numerical algorithm for the computation of this matrix is later given by Decell[1]. An algorithm for the evaluation of the generalized inverse of $A(s) \in R[s]^{r\times m}$ $\forall s \in R$ has later been proposed by Karampetakis[5] while an extension of this algorithm to the evaluation of the generalized inverse of $A(s_1, s_2) \in R[s_1, s_2]^{r\times m}$ $\forall (s_1, s_2) \in C^2$ has also been proposed by the same author (Karampetakis[6]). Both algorithms presented in Karampetakis[5 6] share the same disadvantage, they both depend on the degree of $A(s)$ i.e. q_1 (or the degree of s_1 and s_2 in $A(s_1, s_2)$). More specifically the number of matrices embedded in the evaluation of the generalized inverse of $A(s)$ is analogous to the square of the degree of $A(s)$ i.e. $\sum_{i=1}^{r}(iq_1+1)^2 + 2\times (q_1+1)^2$. Thus in case where for example there is only one big power of s in $A(s) \in R[s]^{2\times 3}$ i.e. s^{80}, and all the other powers of s are zero or one, then the number of matrices used for the evaluation of the generalized inverse of $A(s)$ is about $161^2 + 81^2 + 2\times 81^2 = 45604$ with serious consequences in the speed and accuracy of the algorithm. In that case where big gaps are existing between the powers of s in $A(s)$, we propose in Section 3 an improved algorithm of the one presented in Karampetakis[5] and we extend this algorithm to the two variable case in Section 4.

2 Preliminary Results.

An algorithm for the evaluation of the generalized inverse of $A(s_1, s_2) \in R[s_1, s_2]^{r\times m}$ $\forall (s_1, s_2) \in C^2$ has been proposed in Karampetakis[6]. The reduction of this algorithm in the one variable case is proposed in the sequel :

Algorithm 1. (Computation of the generalized inverse of $A(s)$)
Step 1. Consider the sequences $\{p_1(s), p_2(s), ..., p_r(s)\}$, $\{R_0(s), R_1(s), ..., R_{r-1}(s)\}$ constructed in the following way :

$$\begin{array}{cc} R_0 = I_r & a_1 = -\frac{1}{1} tr[AA^* R_0] \\ R_i = AA^* R_{i-1} + a_i I_r & a_i = -\frac{1}{2} tr[AA^* R_i] \\ i = 1, 2, ..., r-1 & i = 1, 2, ..., r \end{array} \qquad (2)$$

where $(*)$ denotes the conjugate transpose.
Step 2. If $k \neq 0$ is the largest integer such that $p_k(s) \neq 0$ for $s \in L\,(\neq \emptyset) \subseteq C$, then the generalized inverse of $A(s)$ for those $s \in L\,(\neq \emptyset) \subseteq C$ is given by

$$A^{\dagger}(s) = -\frac{A(s)^* R_{k-1}(s)}{p_k(s)} \qquad (3)$$

else ($k = 0$ is the largest integer such that $p_k(s) \neq 0$) $A^{\dagger}(s) = 0$. For those $s \in C - L$ we use the same algorithm. ■

The above algorithm is a symbolical algorithm and can be used in symbolic packages like MAPLE or MATHEMATICA [3]. This algorithm can also be reduced following similar lines with Karampetakis[6] to a three-dimensional numerical algorithm as we can see in what follows.

Algorithm 2. (Computation of the generalized inverse of $A(s)$)
Initialize :

$$R_{0,0,0} = I_r \;\&\; A_{j_1,0} = A_{j_1} \;\&\; A^*_{0,j_1} = A^*_{j_1}$$

Boundary conditions :

$$\begin{array}{c} R_{0,j_1,j_2} = 0 \;\forall j_z > 0 \; z = 1,2 \\ R_{i,j_1,j_2} = 0 \; j_z = iq_1+1, iq_1+2, ..., (r-1)q_1 \\ z = 1,2 \text{ and } i = 0, 1, ..., r-1 \end{array}$$

$$A_{j_1,j_2} = 0 \; \forall j_2 \neq 0 \;\&\; A^*_{j_1,j_2} = 0 \; \forall j_1 \neq 0$$
$$C_{j_1,j_2} = \sum_{n_1=0}^{j_1} \sum_{n_2=0}^{j_2} A_{j_1-n_1,j_2-n_2} A^*_{n_1,n_2}$$
$$j_1 = 0,1,...,q_1 \text{ and } j_2 = 0,1,...,q_1$$

(a) *Recursive relation for*

$$p_i(s) = \sum_{j_1=0}^{iq_1} \sum_{j_2=0}^{iq_2} p_{i,j_1,j_2} s^{j_1} \bar{s}^{j_2}$$

$$p_{i+1,j_1,j_2} =$$
$$= -\frac{1}{(i+1)} \text{trace} \left[\sum_{n_1=0}^{j_1} \sum_{n_2=0}^{j_2} C_{j_1-n_1,j_2-n_2} R_{i,n_1,n_2} \right]$$
$$\text{for } j_1 = 0,1,...,(i+1)q_1, \; j_2 = 0,1,...,(i+1)q_1$$
$$\text{and } i = 0,1,...,r-1$$

(b) *Recursive relation for*

$$R_i(s) = \sum_{j_1=0}^{iq_1} \sum_{j_2=0}^{iq_2} R_{i,j_1,j_2} s^{j_1} \bar{s}^{j_2}$$

$$R_{i+1,j_1,j_2} = \left[\sum_{n_1=0}^{j_1} \sum_{n_2=0}^{j_2} C_{j_1-n_1,j_2-n_2} R_{i,n_1,n_2} \right] +$$
$$+ p_{i+1,j_1,j_2} I_r$$
$$\text{for } k = 0,1,...,(i+1)q_1 \text{ and } i = 0,1,...,r-2$$
$$\text{and } j_1 = 0,1,...,(i+1)q_1, \; j_2 = 0,1,...,(i+1)q_1$$

Terminate :

$$\text{FIND } k : p_{k+1}(s) = p_{k+2}(s) = \cdots = p_r(s) = 0$$
$$\text{or } p_{k+1,j_1,j_2} = p_{k+2,j_1,j_2} = \cdots = p_{r,j_1,j_2} \; \forall j_z \in N$$

Define

$$R_{j_1,j_2} = R_{r-1,j_1,j_2}$$
$$\text{for } j_1 = 0,1,...,(r-1)q_1 \text{ and } j_2 = 0,1,...,(r-1)q_1$$

$$p_{j_1,j_2} = p_{r,j_1,j_2}$$
$$\text{for } j_1 = 0,1,...,(r-1)q_1 \text{ and } j_2 = 0,1,...,(r-1)q_1$$

OUTPUT :

$$A^\dagger(s) = -\frac{A(s)^* R_{k-1}(s)}{p_k(s)} =$$
$$= -\frac{\left(\sum_{j_2=0}^{q_1} A^*_{0,j_2} \bar{s}^{j_2} \right) \left(\sum_{j_1=0}^{(k-1)q_1} \sum_{j_2=0}^{(k-1)q_1} R_{j_1,j_2} s^{j_1} \bar{s}^{j_2} \right)}{\sum_{j_1=0}^{kq_1} \sum_{j_2=0}^{kq_1} p_{j_1,j_2} s^{j_1} \bar{s}^{j_2}} \qquad (4)$$

If $j_2 > j_1$ then
substitute $s^{j_1} \bar{s}^{j_2}$ in (4) for $|s|^{j_1} \bar{s}^{j_2-j_1}$
else
substitute $s^{j_1} \bar{s}^{j_2}$ in (4) for $|s|^{j_2} \bar{s}^{j_1-j_2}$
endif ■

In case however where the degree of $A(s)$ e.g. q_1, is large enough, then in order to execute the above algorithm we need enough a) computer memory so that to keep all the matrices A_i and b) CPU time. Similar problems we have also a) in the evaluation of the inverse of a square one variable polynomial matrix presented by Fragulis[2] or of a square n-variable matrix presented by Karampetakis[4] and b) in the evaluation of the generalized inverse of a non-square two-variable polynomial matrix Karampetakis[6].

Example 1. Consider the polynomial matrix

$$A(s) = \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}}_{A_{80}} s^{80} + \underbrace{\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}}_{A_1} s + \underbrace{\begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{A_0}$$

For the computation of the above polynomial matrix, following the above algorithm, we need to keep in memory the $81^2 = 6561$ matrices A_{j_1,j_2} , the $81^2 = 6561$ matrices C_{j_1,j_2}, the $81^2 = 6561$ matrices R_{1,j_1,j_2} and the $161^2 =: 25921$ products

$$\sum_{n_1=0}^{j_1} \sum_{n_2=0}^{j_2} C_{j_1-n_1,j_2-n_2} R_{1,n_1,n_2}$$

even if most of the above matrices are zero. Obviously the execution time will be analogous to the number of matrices which we use and the rank of $A(s)$. ■

In order to overcome these difficulties we propose in Section 3 an improvement of Algorithm 2, for the computation of the generalized inverse of one-variable polynomial matrices, while in Section 4 we extend this algorithm to the two-variable case.

3 Computation of the generalized inverse of one-variable polynomial matrix.

Define the following "sum" :

$$(\mu_1, \mu_2) \oplus (\nu_1, \nu_2) = (\mu_1 + \nu_1, \mu_2 + \nu_2)$$

and "power" :

$$(s_1, s_2)^{(\nu_1,\nu_2)} = s_1^{\nu_1} \times s_2^{\nu_2}$$

Define also as :

$$\Phi_A = \left\{ \begin{array}{c} (\mu_i, 0) : \text{the set of degrees of} \\ \text{nonzero coefficient matrices of } A(s) \end{array} \right\}$$
$$\Phi_A(i) = \text{the } i\text{th element of } \Phi_A \;\; (\text{let } (\mu_i, 0))$$
$$\bar{\Phi}_A(i) = (0, \mu_i) = \text{the dual of the } i\text{th element of } \Phi_A$$
$$n_A = q = \text{the total number of elements in } \Phi_A$$

Now by setting $s_1 = s$ and $s_2 = \bar{s}$ we can rewrite $A(s)$ as follows :

$$A(s) = \sum_{i=1}^{q} A_{\Phi_A(i)} (s_1, s_2)^{\Phi_A(i)} \tag{5}$$

$$A(s)^* = \sum_{i=1}^{q} A^*_{\Phi_A(i)} (s_1, s_2)^{\bar{\Phi}_A(i)} \tag{6}$$

where (*) denotes the conjugate transpose and $A_{\Phi_A(i)} \neq 0_{r,m}$ $\forall i \in q$. Let also

$$R_i(s) = \sum_{j=1}^{n_i} R_{i,\Phi_i(j)} (s_1, s_2)^{\Phi_i(j)} \tag{7}$$

$$p_i(s) = \sum_{j=1}^{n_i} p_{i,\Phi_i(j)} (s_1, s_2)^{\Phi_i(j)} \tag{8}$$

where

$$\begin{gathered} \Phi_i = \text{the set of degrees of nonzero matrices of } R_i(s) \\ \Phi_i(j) = \text{the } j\text{th element of } \Phi_i \\ n_i = \text{the total number of elements in } \Phi_i \end{gathered}$$

We have that

$$\begin{gathered} A(s)A(s)^* = \\ = \left(\sum_{j=1}^{q} A_{\Phi_A(j)} (s_1, s_2)^{\Phi_A(j)} \right) \times \\ \times \left(\sum_{j=1}^{q} A^*_{\Phi_A(j)} (s_1, s_2)^{\bar{\Phi}_A(j)} \right) = \\ = \sum_{j=1}^{q} \sum_{k=1}^{q} A_{\Phi_A(j)} A^*_{\Phi_A(k)} (s_1, s_2)^{\Phi_A(j) \oplus \bar{\Phi}_A(k)} = \\ = \sum_{j=1}^{n_{AA^*}} C_{\Phi_{AA^*}(j)} (s_1, s_2)^{\Phi_{AA^*}(j)} \end{gathered} \tag{9}$$

where

$$\begin{gathered} \Phi_{AA^*} = \Phi_{AA^*} \cup \left\{ \Phi_A(j) \oplus \bar{\Phi}_A(k) \right\} \\ \text{for } j = 1, 2, ..., q \text{ and } k = q+1, q+2, ..., 2q \end{gathered}$$

and

$$\tilde{n}_{AA^*} = \text{the total number of elements in } \Phi_{AA^*}$$

We subtract in the sequel from Φ_{AA^*} those degrees $\Phi_{AA^*}(j)$ which correspond to zero matrices $C_{\Phi_{AA^*}(j)}$ and we form the new set Φ_{AA^*} with total number of elements, let n_{AA^*}. Then from (9) we have that

$$\begin{gathered} A(s)A(s)^* R_i(s) = \\ = \left(\sum_{j=1}^{n_{AA^*}} C_{\Phi_{AA^*}(j)} (s_1, s_2)^{\Phi_{AA^*}(j)} \right) \times \\ \times \left(\sum_{j=1}^{n_i} R_{i,\Phi_i(j)} (s_1, s_2)^{\Phi_i(j)} \right) = \\ = \sum_{j=1}^{n_{AA^*}} \sum_{k=1}^{n_i} C_{\Phi_{AA^*}(j)} R_{i,\Phi_i(k)} (s_1, s_2)^{\Phi_{AA^*}(j) \oplus \Phi_i(k)} \end{gathered} \tag{10}$$

and thus

$$\begin{gathered} \Phi_{i+1} = \Phi_{i+1} \cup \{ \Phi_{AA^*}(j) \oplus \Phi_i(k) \} \\ \text{for } i = 1, 2, ..., r-1, \; j = 1, 2, ..., n_{AA^*} \\ \text{and } k = 1, 2, ..., n_i \end{gathered}$$

Following similar way with the above product e.g. $A(s)A(s)^*$, we subtract from the set Φ_{i+1} $i = 1, 2, ..., r-1$ those degrees $\Phi_{i+1}(j)$ which corresponds to zero products $C_{\Phi_{AA^*}(j)} R_{\Phi_i(k)}$ and we form the new set Φ_{i+1} with total number of elements n_{i+1} instead of $\tilde{n}_{i+1}$ which the previous Φ_{i+1} had.

Substituting (7), (8) and (10) in the recursive relations (2) we obtain the following recursive algorithm that determines $p_{i+1,\Phi_{i+1}(j)}$ and $R_{i+1,\Phi_{i+1}(j)}$ for $j = 1, 2, ..., n_i$.

Algorithm 3. (Computation of the generalized inverse of $A(s)$)
Initialize :

$$R_{0,(0,0)} = I_m$$

Boundary conditions :

$$\begin{gathered} \Phi_i = \{(0,0)\}, \; n_i = 1 \text{ for } i = 0, 1, ..., r \\ \Phi_A = \{(\mu_1, 0), (\mu_2, 0), ..., (\mu_q, 0)\} = \text{the set} \\ \text{of degrees of nonzero coefficient matrices of } A(s) \\ n_A = q = \text{the total number of elements in } \Phi_A \end{gathered}$$

Main Program
Step 1. Computation of $A(s_1)A(s_2)^*$.
Step 1.1 Computation of a) the coefficient matrix which correspond to the $\Phi_A(j) \oplus \bar{\Phi}_A(k)$–degree of (s_1, s_2) in $A(s_1)A(s_2)^*$ and b) the set Φ_{AA^*} in terms of Φ_A:

$$\begin{gathered} C_{\Phi_A(j) \oplus \bar{\Phi}_A(k)} = C_{\Phi_A(j) \oplus \bar{\Phi}_A(k)} + A_{\Phi_A(j)} A^*_{\Phi_A(k)} \\ \Phi_{AA^*} = \Phi_{AA^*} \cup \left\{ \Phi_A(j) \oplus \bar{\Phi}_A(k) \right\} \end{gathered}$$

for $j = 1, 2, ..., n_A$ and $k = 1, 2, ..., n_A$.
Step 1.2 Computation of the total number of elements in Φ_{AA^*}

$$\tilde{n}_{AA^*} = \text{the total number of elements in } \Phi_{AA^*}$$

Step 1.3 Set $s = 0$ and apply for $j = 1, 2, ..., \tilde{n}_{AA^*}$

$$\begin{gathered} \text{If } C_{\Phi_{AA^*}(j)} = 0 \text{ then } \Phi_{AA^*} = \Phi_{AA^*} - \{\Phi_{AA^*}(j)\} \\ \text{and } s = s + 1 \end{gathered}$$

Step 1.4 Set $n_{AA^*} = \tilde{n}_{AA^*} - s$.
Step 2. Apply for $i = 0, 1, 2, .., r-1$ the following steps :
Step 2.1. Computation of a) the coefficient matrix which correspond to the $\Phi_{AA^*}(j) \oplus \Phi_i(k)$–degree of (s_1, s_2) in

$A(s_1)A(s_2)^* R_i(s_1,s_2)$ and b) the set Φ_{i+1} in terms of Φ_{AA^*} and Φ_i

$$Q_{\Phi_{AA^*}(j)\oplus\Phi_i(k)} = Q_{\Phi_{AA^*}(j)\oplus\Phi_i(k)} + C_{\Phi_{AA^*}(j)} R_{i,\Phi_i(k)}$$
$$\Phi_{i+1} = \Phi_{i+1} \cup \{\Phi_{AA^*}(j) \oplus \Phi_i(k)\}$$

for $j = 1,2,...,n_{AA^*}$ and $k = 1,2,...,n_i$

Step 2.2. Computation of the total number of elements in Φ_{i+1}

$$\tilde{n}_{i+1} = \text{total number of elements in } \Phi_{i+1}$$

Step 2.3. Computation of $p_{i+1,\Phi_{i+1}(j)}$ and $R_{i+1,\Phi_{i+1}(j)}$:

Set $s = 0$

For $j = 1,2,..,\tilde{n}_{i+1}$

If $Q_{\Phi_{i+1}(j)} = 0$ then

$$\Phi_{i+1} = \Phi_{i+1} - \{\Phi_{i+1}(j)\} \text{ and } s = s+1$$

else

$$p_{i+1,\Phi_{i+1}(j)} = -\tfrac{1}{i+1} tr\left[Q_{\Phi_{i+1}(j)}\right]$$

If $i < r-1$ then

$$R_{i+1,\Phi_{i+1}(j)} = Q_{\Phi_{i+1}(j)} + p_{i+1,\Phi_{i+1}(j)} I_r$$

Set $Q_{\Phi_{i+1}(j)} = 0$

endif

endif

Next j

$n_{i+1} = \tilde{n}_{i+1} - s$

Terminate : FIND k :

$$p_{k+1,\Phi_{k+1}(j)} = p_{k+2,\Phi_{k+2}(j)} = \cdots = p_{r,\Phi_r(j)} = 0$$

WHILE $\exists j : p_{k,\Phi_k(j)} \neq 0$ define

$$R_{\Phi_{k-1}(i)} = R_{k-1,\Phi_{k-1}(i)} \text{ for } i = 1,...,n_{k-1}$$
$$p_{\Phi_k(i)} = p_{k,\Phi_k(i)} \text{ for } i = 1,...,n_k$$

OUTPUT : The generalized inverse of $A(s)$ will be

$$A^\dagger(s_1,s_2) = -\frac{A(s_1,s_2)^* R_{k-1}(s_1,s_2)}{p_k(s_1,s_2)} = -\frac{\left(\sum_{j=1}^{q} A^*_{\Phi_A(j)}(s_1,s_2)^{\Phi_A(j)}\right)}{\left(\sum_{j=1}^{n_k} p_{\Phi_k(j)}(s_1,s_2)^{\Phi_k(j)}\right)} \times \left(\sum_{j=1}^{n_{k-1}} R_{\Phi_{k-1}(j)}\,(s_1,s_2)^{\Phi_{k-1}(j)}\right) \tag{11}$$

for those $s \in L (\neq \emptyset) : p_k(s) \neq 0$.

If $\Phi_{k-1}(j) = (\mu_j,\nu_j)$ and $\mu_j > \nu_j$ then substitute $(s_1,s_2)^{\Phi_{k-1}(j)}$ in (11) for $|s|^{\nu_j} \, \bar{s}^{\mu_j-\nu_j}$

else

substitute $(s_1,s_2)^{\Phi_{k-1}(j)}$ in (11) for $|s|^{\mu_j} \, \bar{s}^{\nu_j-\mu_j}$

endif

For those $s \in C - L$ we use the same algorithm. ■

It is easily seen that the above algorithm use in computations only the nonzero coefficient matrices of $A(s)$. This is a big advantage in cases where the degrees involved on $A(s)$ are big enough or they have big enough gaps between each other. In all other cases the proposed Algorithm 2, is better, since no extra controls are needed for the set of degrees of nonzero coefficient matrices of $A(s)R_i(s)$. Actually the above algorithm uses n_A matrices of the form $A_{\Phi_A(j)}$, $\tilde{n}_{AA^*}$ matrices of the form $C_{\Phi_{AA^*}(j)}$, $\max\left\{\tilde{n}_i, i = 1,2,..,r\right\}$ matrices of the form $Q_{\Phi_i(j)}$ and $\sum_{i=1}^{r-1} \tilde{n}_i$ matrices of the form $R_{i,\Phi_i(j)}$ and thus totally

$$n_A + \tilde{n}_{AA^*} + \max\left\{\tilde{n}_i, i = 1,2,..,r\right\} + \sum_{i=1}^{r-1} \tilde{n}_i \text{ matrices}$$

The above algorithm remains the same in case where the polynomial matrix $A(s) \in C[s]^{r\times m}$ e.g. the only change it needs is in the definition of the set Φ_A. An extension of the above algorithm in the 2-D case is proposed in the following section.

4 Computation of the generalized inverse of a two-variable polynomial matrix.

Define the following "sum" :

$$(\mu_1,\mu_2,\mu_3,\mu_4) \oplus (\nu_1,\nu_2,\nu_3,\nu_4) = = (\mu_1+\nu_1, \mu_2+\nu_2, \mu_3+\nu_3, \mu_4+\nu_4)$$

and "power" :

$$(s_1,s_2,s_3,s_4)^{(\nu_1,\nu_2,\nu_3,\nu_4)} = s_1^{\nu_1} \times s_2^{\nu_2} \times s_3^{\nu_3} \times s_4^{\nu_4}$$

Define also as :

$$\Phi_A = \left\{ \begin{array}{c} (\mu_i,\nu_i,0,0) : \text{the set of degrees of} \\ \text{nonzero coefficient matrices of } A(s_1,s_2) \end{array} \right\}$$
$$\Phi_A\,(i) = \text{the } i\text{th element of } \Phi_A \text{ (let } (\mu_i,\nu_i,0,0))$$
$$\bar{\Phi}_A\,(i) = (0,0,\mu_i,\nu_i) = = \text{the dual of the } i\text{th element of } \Phi_A$$
$$n_A = q = \text{the total number of elements in } \Phi_A$$

Now by setting $s_3 = \bar{s}_1$ and $s_4 = \bar{s}_2$ we can rewrite $A(s_1,s_2)$ as follows :

$$A(s_1,s_2) = \sum_{i=1}^{q} A_{\Phi_A(i)}\,(s_1,s_2,s_3,s_4)^{\Phi_A(i)} \tag{12}$$

$$A(s_1,s_2)^* = \sum_{i=1}^{q} A^*_{\Phi_A(i)}(s_1,s_2,s_3,s_4)^{\bar{\Phi}_A(i)} \qquad (13)$$

where (*) denotes the conjugate transpose and $A_{\Phi_A(i)} \neq 0_{r,m}$ $\forall i \in q$. Then applying the same techniques with the ones described above we get the following algorithm for the computation of the inverse of $A(s_1,s_2)$:

Algorithm 4. (Computation of the generalized inverse of $A(s_1,s_2)$)
Initialize :

$$R_{0,(0,0,0,0)} = I_m$$

Boundary conditions :

$$\Phi_i = \{(0,0,0,0)\},\ n_i = 1 \text{ for } i = 0,1,...,r$$
$$\Phi_A = \{(\mu_1,\nu_1,0,0),(\mu_2,\nu_2,0,0),...,(\mu_q,\nu_q,0,0)\} =$$
$$= \text{the set of degrees of nonzero coefficient matrices of } A(s)$$
$$n_A = q = \text{the total number of elements in } \Phi_A$$

Main Program
Step 1. Computation of $A(s_1,s_2)A(s_3,s_4)^*$.
Step 1.1 Computation of a) the coefficient matrix which correspond to the $\Phi_A(j)\oplus\ \bar{\Phi}_A\ (k)$−degree of (s_1,s_2,s_3,s_4) in $A(s_1,s_2)A(s_3,s_4)^*$ and b) the set Φ_{AA^*} in terms of Φ_A:

$$C_{\Phi_A(j)\oplus\bar{\Phi}_A(k)} = C_{\Phi_A(j)\oplus\bar{\Phi}_A(k)} + A_{\Phi_A(j)}A^*_{\Phi_A(k)}$$
$$\Phi_{AA^*} = \Phi_{AA^*} \cup \left\{\Phi_A(j)\oplus\ \bar{\Phi}_A\ (k)\right\}$$

for $j = 1,2,...,n_A$ and $k = 1,2,...,n_A$.
Step 1.2 Computation of the total number of elements in Φ_{AA^*}

$$\tilde{n}_{AA^*} = \text{the total number of elements in } \Phi_{AA^*}$$

Step 1.3 Set $s=0$ and apply for $j = 1,2,...,\tilde{n}_{AA^*}$

$$\text{If } C_{\Phi_{AA^*}(j)} = 0 \text{ then } \Phi_{AA^*} = \Phi_{AA^*} - \{\Phi_{AA^*}(j)\} \text{ and } s = s+1$$

Step 1.4 $n_{AA^*} = \tilde{n}_{AA^*} - s$
Step 2. Apply for $i = 0,1,2,...,r-1$ the following steps
Step 2.1. Computation of a) the coefficient matrix which correspond to the $\Phi_{AA^*}(j)\oplus\Phi_i(k)$−degree of (s_1,s_2) in $A(s_1,s_2)A(s_3,s_4)^*R_i(s_1,s_2,s_3,s_4)$ and b) the set Φ_{i+1} in terms of Φ_{AA^*} and Φ_i

$$Q_{\Phi_{AA^*}(j)\oplus\Phi_i(k)} = Q_{\Phi_{AA^*}(j)\oplus\Phi_i(k)} + C_{\Phi_{AA^*}(j)}R_{i,\Phi_i(k)}$$
$$\Phi_{i+1} = \Phi_{i+1} \cup \{\Phi_{AA^*}(j)\oplus\Phi_i(k)\}$$

for $j = 1,2,...,n_{AA^*}$ and $k = 1,2,...,n_i$
Step 2.2. Computation of the total number of elements in Φ_{i+1}

$$\tilde{n}_{i+1} = \text{total number of elements in } \Phi_{i+1}$$

Step 2.3. Computation of $p_{i+1,\Phi_{i+1}(j)}$ and $R_{i+1,\Phi_{i+1}(j)}$:
Set $s=0$
For $j = 1,2,..,n_{i+!}$
If $Q_{\Phi_{i+1}(j)} = 0$ then

$$\Phi_{i+1} = \Phi_{i+1} - \{\Phi_{i+1}(j)\} \text{ and } s = s+1$$

else

$$p_{i+1,\Phi_{i+1}(j)} = -\frac{1}{i+1}tr\left[Q_{\Phi_{i+1}(j)}\right]$$
If $i < r-1$ then
$$R_{i+1,\Phi_{i+1}(j)} = Q_{\Phi_{i+1}(j)} + p_{i+1,\Phi_{i+1}(j)}I_r$$
Set $Q_{\Phi_{i+1}(j)} = 0$
endif

endif
Next j
$n_{i+1} = \tilde{n}_{i+1} - s$
Terminate : FIND k :

$$p_{k+1,\Phi_{k+1}(j)} = p_{k+2,\Phi_{k+2}(j)} = \cdots = p_{r,\Phi_r(j)} = 0$$

WHILE $\exists j : p_{k,\Phi_k(j)} \neq 0$ define

$$R_{\Phi_{k-1}(i)} = R_{k-1,\Phi_{k-1}(i)} \text{ for } i = 1,...,n_{k-1}$$
$$p_{\Phi_k(i)} = p_{k,\Phi_k(i)} \text{ for } i = 1,...,n_k$$

OUTPUT : The generalized inverse of $A(s_1,s_2)$ will be

$$A^\dagger(s_1,s_2,s_3,s_4) = -\frac{A(s_1,s_2,s_3,s_4)^* R_{k-1}(s_1,s_2,s_3,s_4)}{p_k(s_1,s_2,s_3,s_4)} = -\frac{\left(\sum_{j=1}^{q} A^*_{\Phi_A(j)}(s_1,s_2,s_3,s_4)^{\bar{\Phi}_A(j)}\right)}{\left(\sum_{j=1}^{n_k} p_{\Phi_k(j)}(s_1,s_2,s_3,s_4)^{\Phi_k(j)}\right)} \times \left(\sum_{j=1}^{n_{k-1}} R_{\Phi_{k-1}(j)}(s_1,s_2,s_3,s_4)^{\Phi_{k-1}(j)}\right) \qquad (14)$$

for those $(s_1,s_2) \in L\ (\neq \emptyset) : p_k(s_1,s_2) \neq 0$.
If $\Phi_{k-1}(j) = (\mu_1,\mu_2,\mu_3,\mu_4)$ then
If $\mu_3 > \mu_1$ then
substitute $s_1^{\mu_1} \times s_3^{\mu_3}$ in (14) for $|s_1|^{\mu_1}\ \bar{s}_1^{\ \mu_3-\mu_1}$
else
substitute $s_1^{\mu_1} \times s_3^{\mu_3}$ in (14) for $|s_1|^{\mu_3}\ \bar{s}_1^{\ \mu_1-\mu_3}$
endif
If $\mu_4 > \mu_2$ then
substitute $s_2^{\mu_2} \times s_4^{\mu_4}$ in (14) for $|s_2|^{\mu_2}\ \bar{s}_2^{\ \mu_4-\mu_2}$

else
substitute $s_2^{\mu_2} \times s_4^{\mu_4}$ in (14) for $|s_2|^{\mu_4} \bar{s_2}^{\mu_2-\mu_4}$
endif
endif
For those $s \in C - L$ we use the same algorithm. ■

The algorithm remains the same for the case where $A(s_1, s_2) \in C[s_1, s_2]^{r\times m}$ with the only change in the definition of Φ_A in Step 1. It is easily seen that the total number of matrices used by the above algorithm is :

$$n_A + \tilde{n}_{AA^*} + \max\left\{\tilde{n}_i, i = 1, 2, .., r\right\} + \sum_{i=1}^{r-1} \tilde{n}_i$$

Corollary 1. The number of matrices used for the evaluation of the generalized inverse of $A(s)$ in Example 1, by using Algorithm 3, is

$$n_A + \tilde{n}_{AA^*} + \max\left\{\tilde{n}_i, i = 1, 2, .., r\right\} + \sum_{i=1}^{r-1} \tilde{n}_i = \\ = 3 + 9 + \max\{5, 14\} + 5 = 31$$

instead of $81^2 + 81^2 + 81^2 + 161^2 = 45604$ matrices by using Algorithm 2. ■

Similar algorithms can be applied for the evaluation of the generalized inverse of a polynomial matrix with more variables.

5 Conclusions

Two recursive algorithms for the evaluation of the generalized inverse of one-variable and two-variable nonsquare polynomial matrices have been evaluated. These algorithms consists improvements of the corresponding algorithms presented in Karampetakis[5 6]. The whole theory has been illustrated via an example. The above results may also be extended to the nth-variable case where $n > 2$ while the evaluation of the Laurent expansion of the generalized inverse on both cases is under research.

Acknowledgments
This work was supported by the Greek National Foundation.

References

1. Decell H.P., 1965, An application of the Cayley-Hamilton theorem to generalized matrix inversion, *SIAM Review*, Vol.**7**, pp.526-528.
2. Fragulis G., Mertzios B.G. and Vardulakis A.I.G., 1991, Computation of the inverse of a polynomial matrix and evaluation of its Laurent expansion, *Int. J. Control*, Vol.**53**, pp.431-443.
3. Jones J., Karampetakis N.P. and Pugh A.C., 1996, Computation of the generalized inverse of a rational matrix via MAPLE and applications, *Proceedings of the CACSD-96*.
4. Karampetakis N.P., Mertzios B.G. and Vardulakis A.I.G., 1994, Computation of the transfer function matrix and its Laurent expansion of generalized two-dimensional systems, *Int. J. Control*, Vol.**60**, pp.521-541.
5. Karampetakis N.P., 1997a, Computation of the generalized inverse of a polynomial matrix and applications., *Linear Algebra and its Applications*, Vol. **252**, pp.35-60.
6. Karampetakis N.P., 1997b, Generalized inverses of two variable polynomial matrices and applications,*Circuit Systems & Signal Process*, Vo.**16**, No.4, pp.439-453.
7. Kucera V., 1993, Diophantine equations in Control - A survey, Automatica, Vol.29, pp.1361-1375.
8. Lovass Nagy V., Miller R. and Powers D., 1976, Research Note : On the application of matrix generalized inverses to the construction of inverse systems., *Int. J. Control*, Vol.**24**, pp.733-739.
9. Lovass Nagy V., Miller R. and Powers D., 1978, Transfer function matrix synthesis by matrix generalized inverses, *Int. J. Control*, Vol.**27**, pp.387-391.
10. Lovass Nagy V., Miller R. and Powers D., 1978, An introduction to the application of the simplest matrix generalized inverse in system science, *IEEE Trans. on Auto. Control*, Vol.**25**, pp.766-771.
11. Penrose R., 1955, A generalized inverse for matrices, *Proc. Cambridge Philos. Soc.*, Vol.**51**, pp.406-413.